Design and Analysis of Algorithms Assignment 1

Harrison Lee

January 21, 2019

Question 1. *"Bottleneck" Nodes in a Graph*

Claim: Given two nodes, s and t, in n-node indirected graph $G = (V, E)$ with a distance greater than $n/2$, there exists some node v not equal to either s or t that, when deleted, destroys all paths from s to t.

Proof. A path of distance greater than $n/2$ takes $n/2 + 1$ nodes at least. Excluding s and t this is $(n - 2)/2 + 1 = n/2$ nodes. Let's call this path "path A".

For node v to be deletable without destroying the other path, path B, v cannot be in B.

B must also have a distance greater than $n/2$ to maintain s and ts' distance.

B cannot share nodes with A other than s and t and requires $n/2$ nodes unique from A.

There are only $n - 2$ non-s or non-t nodes, but A and B require a total of n unique nodes, so B cannot exist.

□

Algorithm 1. *Find node v*

Begin with a Depth-First-Search, as written in the textbook. Use it to find the shortest path from s to t.

Mark all nodes discovered in that shortest path as "Used", numbering them by their distance to t.

Repeat Depth-First-Search, starting from s, but do not traverse past "Used" nodes. Instead, mark them as "Re-Found". Save whichever "Re-Found" node that is closest to t.

Once Depth-First-Search fails and cannot continue, return the saved "Re-Found" node as v.

Proof. Prove this algorithm is $O(n + m)$:

Depth-First-Search is known to be $O(n + m)$, as noted in the textbook. Each edge and node is traversed at most once.

This algorithm conducts Depth-First-Search twice.

This algorithm is $O(2n + 2m)$, which is close enough to $O(n + m)$ for our purposes.

□

Question 2.

Claim: $P(n) : n = p_1 * p_2 * \ldots p_k \forall n > 1$, where p is a prime number.

Proof. **Base Case:** $P(2) = 2 * 1$

Assume $P(1) \wedge P(2) \wedge \ldots \wedge P(n)$

Prove $P(1) \wedge P(2) \wedge \ldots \wedge P(n) \rightarrow P(n+1)$

Case 1: If $n+1$ is a prime number, it is divisible by itself and 1, so $P(n+1) = T$ if $n+1$ is prime.

Case 2: Otherwise, if $n+1$ is not prime, then $n+1 = x * y$.

$x, y < n$ and $P(x), P(y) = T$, each with their own factors.

By recursively finding the factors of $P(x)$ and $P(y)$, which would then be factors to $P(n)$ as well.

Therefore $P(n+1)$ is true, and $P(n) = T \forall n > 1$.

□

Question 3.

Claim: $\sum_{i=1}^{n} \frac{1}{\sqrt{i}} \geq \sqrt{n}$

Proof. **Base Case:** $P(1) : 1 \geq \sqrt{1}$

Assume $\sum_{i=1}^{n} \frac{1}{\sqrt{i}} \geq \sqrt{n}$

Prove $P(n) \rightarrow P(n+1)$**:** Suppose $P(n)$ is true for $n = k$. When $n = k+1$ we have.

$\sum_{i=1}^{n+1} \frac{1}{\sqrt{i}} \geq \sqrt{n+1}$

$\sum_{i=1}^{n} \frac{1}{\sqrt{i}} + \frac{1}{\sqrt{n+1}} \geq \sqrt{n+1}$

$\sum_{i=1}^{n} \frac{1}{\sqrt{i}} * \sqrt{n+1} + 1 \geq n+1$

$\sum_{i=1}^{n} \frac{1}{\sqrt{i}} * \sqrt{n+1} \geq n$

$\sum_{i=1}^{n} \frac{1}{\sqrt{i}} * \sqrt{n+1} \geq \sqrt{n} * \sqrt{n+1} \geq n$

$\sqrt{n} * \sqrt{n+1} \geq \sqrt{n} * \sqrt{n}$

$\sqrt{n+1} \geq \sqrt{n}$

Therefore, by induction, $P(n+1)$ is true $\forall n \geq 1$.

□

Question 4.

Claim: $x^n + \frac{1}{x^n}$ is an integer, given $x + \frac{1}{x}$ is an integer and $\forall n \geq 1$

Proof. **Base Case:** $P(1) : x^1 + \frac{1}{x^1} \in \mathbb{Z}$; $P(0) : x^0 + \frac{1}{x^0} = 1 + 1 \in \mathbb{Z}$

Assume $x^n + \frac{1}{x^n} \in \mathbb{Z}$ is true for $n = 1$ and $n = 0$

Prove $P(n) \wedge P(n-1) \wedge P(1) \rightarrow P(n+1)$**:** Assume $P(n)$ and $P(n-1)$. Prove for all other cases by induction.

$x^{n+1} + \dfrac{1}{x^{n+1}}$

$= x * x^n + \dfrac{1}{x * x^n}$

$= (x^n + \dfrac{1}{x^n}) * (x + \dfrac{1}{x}) - (x^{n-1} + \dfrac{1}{x^{n-1}})$

Explanation: $\{\ (x^n + \dfrac{1}{x^n}) * (x + \dfrac{1}{x}) = (x^{n+1} + \dfrac{1}{x^{n+1}}) + (x^{n-1} + \dfrac{1}{x^{n-1}})\ \}$

Integers multiplied together or subtracted from by an integer creates another integer.

Therefore $P(n)$ is true for all n

□

Question 5.

Here is a definition of a round-robin tournament.

(1) Base Case: $P(2)$: Two nodes are connected by one arrow, a win or a loss.

(2) Constructor Rule: When a node is added, it is given either a win or a loss with every other node.

P(n): A network with n nodes in our recursive definition can be sorted linearly so that every node lost to the node ahead of it while winning to the one before it.

Proof. **Base Case:** P(2): The two nodes can be sorted, with the one that won in the front of the line while the other node is behind it.

Prove $P(n) \rightarrow P(n+1)$**:** Assume $P(n) = T$, with a working linear order of nodes.

<u>Case 1:</u> The newly added node has only wins.

It can be placed in front of all other nodes in the line. $P(n)$ will define the order of all other nodes in the line behind the new node.

<u>Case 2:</u> The newly added node has only losses.

It can be placed behind all other nodes in the $P(n)$ line.

<u>Case 3:</u> The new node has both wins and losses.

Temporarily ignore one of the nodes with the most losses.

The remaining nodes can create a $P(n)$ line. Pick a line where the removed node lost to the last node.

The node with the most losses can be added to the end of this line.

Therefore, by induction, $P(n) = T \forall n \in \mathbb{N}$

□